AFRL-RH-BR-TR-2008-0023

Computation and Modeling for Laser Propagation in Ocular Tissues

Dhiraj K. Sardar
Raylon M. Yow
Deepthi Takkalapally

University of Texas at San Antonio

Robert J. Thomas

Air Force Research Laboratory

November 2007
Final Report for July 2006 – November 2007

<u>DESTRUCTION NOTICE</u> – Destroy by any method that will prevent disclosure of contents or reconstruction of this document.

Air Force Research Laboratory
Human Effectiveness Directorate
Directed Energy Bioeffects Division
Optical Radiation Branch
Brooks City-Base, TX 78235

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U. S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the Human Systems Wing (HSW/PA) Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RH-BR-TR-2008-0023 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

__________//SIGNED//____________
ALAN J. RICE, LT
Contract Representative

_________//SIGNED//_____________
GARRETT D. POLHAMUS, PhD, DR-IV
Chief, Directed Energy Bioeffects Division

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE			*Form Approved* *OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**			
1. REPORT DATE *(DD-MM-YYYY)* November 2007	**2. REPORT TYPE** Final Technical Report		**3. DATES COVERED** *(From - To)* July 2006 – November 2007
4. TITLE AND SUBTITLE **COMPUTATION AND MODELING FOR LASER PROPAGATION IN OCULAR TISSUES**			**5a. CONTRACT NUMBER**
			5b. GRANT NUMBER HBCU/MI FA8650-06-1-6746
			5c. PROGRAM ELEMENT NUMBER 62202F
6. AUTHOR(S) *Dhiraj K. Sardar ; *Raylon M. Yow ; *Deepthi Takkalapally; ^Robert J. Thomas			**5d. PROJECT NUMBER** 7757
			5e. TASK NUMBER B2
			5f. WORK UNIT NUMBER 32
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) *University of Texas at San Antonio, College of Sciences Department of Physics, One UTSA Circle, San Antonio, TX 78249-1644; ^Air Force Research Laboratory, Human Effectiveness Directorate, Directed Energy Bioeffects Division, Optical Radiation Branch, 2624 Louis Bauer Dr., Brooks City-Base, TX 78235-5128			**8. PERFORMING ORGANIZATION REPORT NUMBER** AFRL-RH-BR-TR-2008-0023
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Materiel Command, Air Force Research Laboratory, Human Effectiveness Directorate, Directed Energy Bioeffects Division, Optical Radiation Branch, 2624 Louis Bauer Dr., Brooks City-Base, TX 78235-5128			**10. SPONSOR/MONITOR'S ACRONYM(S)**
			11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-RH-BR-TR-2008-0023
12. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited, Public Affairs Case File No. 08-118, 1 May 2008.			
13. SUPPLEMENTARY NOTES Contract Monitor – Dr. Robert Thomas			

14. ABSTRACT

A computational model for the propagation of laser radiation within cylindrical geometry is developed in C++. This model employs a finite-difference technique to model the (2,2) Padé approximant of the light propagation (Helmholtz) operator to solve the scalar Helmholtz equation obtained using the slowly-varying envelope formalism. This technique is capable of handling wide-angle propagation and refractive index variation while still maintaining numerical speed and simplicity. In addition, this model uses a non-linear map from the infinite physical space to a finite computational space to avoid spurious reflections from the computational window edge and improve computational efficiency. Also, the model depends solely on the spatial refractive index and hence can be coupled in a time-slicing scheme to an optical-thermal model that can include linear and non-linear optical effects as well as capture thermal lensing. As an application of the model developed here, predicted irradiance at the retina of laser light incident on the human eye could be used to establish new maximum permissible exposure (MPE) limits.

15. SUBJECT TERMS:
Beam Propagation, Padé Approximant, Finite-Difference

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Robert J. Thomas
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	SAR	28	**19b. TELEPHONE NUMBER** *(include area code)*

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

This page intentionally left blank

Table of Contents

Acknowledgments

This work was supported by the United States Air Force Research Laboratory Human Effectiveness Directorate, Wright-Patterson Base, Dayton, Ohio grant HBCU/MI FA8650-06-1-6746.

1. Introduction

Quantitative understanding of laser-tissue interaction requires a thorough knowledge of the distribution and propagation of light intensity in biological media. This is particularly important when determining the distribution of light within ocular tissue and specifically in determining the irradiance at the retina to a beam of light incident on the eye. The light distribution intensity can be obtained from the solution of the Helmholtz equation using the slowly-varying envelope formalism.[1] The beam-propagation method is at present the most widely used tool employed in this study. The paraxial (Fresnel) approximation is especially popular largely owing to its numerical speed and simplicity.

However, the paraxial approximation severely limits the formalism in the following two respects. First, beams containing appreciable Fourier components at angles of more than a few degrees from the propagation axis will experience substantial phase errors; thus the method cannot treat wide-angle propagation. Second, beams propagating through regions with indices of refraction that differ by more than a few percent from the input reference index will also suffer serious phase distortion.[1]

The Padé approximant scheme, which includes the paraxial approximation, eliminates some of the error in the paraxial beam propagation model. The Padé approximant scheme is also computationally efficient, as compared to an approach based on the method of lines.[1] In this report, we develop an application of the finite-difference method to the Padé approximant formalism for optical beam propagation. This approach offers substantial improvements in both wide-angle propagation and index variation tolerance while incurring (in the two-dimensional case) only a modest numerical penalty.[1] Also, employing a non-linear map from a infinite physical space to an finite computational space avoids spurious reflections from the computational window edge and improves computational efficiency.

In addition, a model sufficient enough to accurately describe laser propagation in biological media must also address time-dependent optical-thermal effects including thermal lensing as well as linear and non-linear optical effects. Thermal lensing is the gradient of the refractive index caused by nonuniform heating. Linear optical effects include a linear correlation between the absorption and scattering coefficients and the temperature of the medium. Non-linear optical effects include non-linear absorption, which captures the dependence of the absorption coefficient on the intensity of the incident light. Motamedi, et al.[2] and Lin, et al.[3] have shown that biological materials (e.g. tissue and blood) possess strong thermal lensing effects. Walsh and Cummings[4] report that moderate transient changes of the absorption coefficient are observed in IR-laser ablation at wavelengths around 3 μm (i.e. close to the absorption peak of tissue water). Lin et al.[5] suggest that the measured transmittance and reflectance decrease and increase, respectively as the temperature-dependent scattering coefficient increases. Vodopyanov[6] shows that the rise of the water temperature during the ablation process can induce a decrease of the absorption coefficient by as much as one order of magnitude. Stolarski, et al.[7] have reported non-linear absorption effects of melanin-doped samples; these

studies are of particular interest to laser bioeffects in the retina, for which melanin plays a key role in the absorption of laser energy.

Moreover, these effects are interrelated. A sample illuminated with light will absorb some of the energy as quantified by the local intensity of light and local absorption coefficient, which in turn depends on the local temperature and the local recent history of the intensity. The absorbed energy causes a temperature rise that induces a thermal lensing effect. This thermal lensing effect along with the temperature-dependent scattering coefficient changes the distribution of the light. Also, Vodopyanov[6] shows that the increase in temperature of water, a large component of most tissues, is associated with a shift of the absorption peak towards shorter wavelengths. These interdependencies of temperature and light distribution, though seemingly complex, are capable of being modeled by a numerical beam-propagation model coupled to a time-dependent optical-thermal model. Both models can be developed independently and coupled in a time-slicing scheme. Moreover, a time-independent beam-propagation model depends only on the spatial refractive index of the sample. Therefore, the specific aim of this report was to develop a time-independent finite-difference model of light distribution in cylindrical geometry that can be coupled to a time-dependent optical-thermal model.

As an application of the above mentioned model, the irradiance at the retina from laser light entering the eye can be predicted. Upon verification of the model, possibly with a z-scan, the predicted incidence light at the retina could be used to establish new maximum permissible exposure (MPE) limits.

2. Non-Paraxial Scalar Wave Equation

Consider the scalar Helmholtz equation in cylindrical geometry obtained by using the slowly-varying envelope formalism:

$$\frac{1}{r}\frac{\partial}{\partial r}\left[\frac{\partial H(r,z)}{\partial r}\right]+2ik\frac{\partial H(r,z)}{\partial z}+\frac{\partial^2 H(r,z)}{\partial z^2}+k_0^2\left[n^2(\vec{r})-n_0^2\right]H(r,z)=0 \qquad (1)$$

where $k = k_0 n_0$ with k_0 the free-space propagation constant, n_0 the (input) reference refractive index. We map the radial coordinate r of infinite support $[0,\infty]$ to a variable of finite support $[0,\pi/2]$ through the mapping function

$$r = \eta \tan\rho \qquad (2)$$

Where η is the coordinate scaling parameter that allows us to control the sampling. Such a representation maps onto the entire physical space, eliminating boundary conditions that could lead to spurious reflections, as well creating a uniform grid from a non-

uniform grid such that a finer grid spacing is placed around the origin.[8,9] The latter is quite useful for modeling beam propagation with cylindrical symmetry about the origin.

Substitution of Eq. (2) into Eq. (1) yields

$$\frac{\partial H}{\partial z} - \frac{i}{2k}\frac{\partial^2 H}{\partial z^2} = \frac{iP}{2k}H \tag{3}$$

where the operator P is defined by:

$$P \equiv k_0^2[n^2(\vec{r}) - n_0^2] + \frac{\cos^4\rho}{\eta^2\sin\rho}(2\cos^2\rho - 1)\frac{\partial}{\partial\rho} \quad . \tag{4}$$

Formally rewriting Eq. (3) in the form:

$$\frac{\partial H}{\partial z} = \frac{\frac{iP}{2k}}{1 - \frac{i}{2k}\frac{\partial}{\partial z}} H \tag{5}$$

suggests the recursion:

$$\left.\frac{\partial}{\partial z}\right|_n H = \frac{\frac{iP}{2k}}{1 - \frac{i}{2k}\frac{\partial}{\partial z}} H \quad . \tag{6}$$

With $\left.\frac{\partial}{\partial z}\right|_0 = 0$, we arrive at the expression:

$$\frac{\partial H}{\partial z} \approx \left.\frac{\partial}{\partial z}\right|_4 H = \frac{\frac{iP}{2k} + \frac{iP^2}{4k^3}}{1 + \frac{3P}{4k^2} + \frac{P^2}{16k^4}} H \quad . \tag{7}$$

Equation (7) is the (2,2) Padé approximant for the Helmholtz operator. Hadley[1] shows the relative phase error incurred by the use of the (2,2) Padé approximate operator is less than 0.2 for up to 65° for an incident plane wave.

3. Finite-Difference Model

Discretization

Finite-difference equations may be derived from Eq. (7) yielding:

$$D(H_j^{m+1} - H_j^m) = \frac{i\Delta z}{2} N(H_j^m + H_j^{m+1}) \tag{8}$$

where N and D are the numerator and denominator, respectively, of Eq. (7). The superscript in Eq. (8) designates the z position while the subscript designates the radial position. Substituting for N and D for the numerator and denominator in Eq. (8), respectively, from Eq. (7) yields:

$$H_j^{m+1} + \alpha P H_j^{m+1} + \beta P^2 H_j^{m+1} = H_j^m + \gamma P H_j^m + \delta P^2 H_j^m \tag{9}$$

where

$$\alpha = \frac{3}{4k^2} - \frac{i\Delta z}{4k} \qquad \beta = \frac{1}{16k^4} - \frac{i\Delta z}{8k^3} \ . \tag{10}$$

The use of centered spatial differencing results in the following form for the operator P for the axially symmetric case:

$$PH_j^m = A_j H_{j-1}^m + B_j^m H_j^m + C_j H_{j+1}^m \tag{11}$$

and

$$\begin{aligned} P^2 H_j^m = {} & A_j (A_{j-1} H_{j-2}^m + B_{j-1}^m H_{j-1}^m + C_{j-1} H_j^m) \\ & + B_j^m (A_j H_{j-1}^m + B_j^m H_j^m + C_j H_{j+1}^m) \\ & + C_j (A_{j+1} H_j^m + B_{j+1}^m H_{j+1}^m + C_{j+1} H_{j+2}^m) \end{aligned} \tag{12}$$

where

$$A_j = \frac{\cos^4 \rho_j}{\eta^2} \frac{1}{(\Delta\rho)^2} - \frac{\cos^3 \rho_j}{\eta^2 \sin \rho_j} \frac{1}{2(\Delta\rho)^2} (2\cos^2 \rho_j - 1)$$

$$B_j^m = k_0^2\left[n^2\left(\rho_j, z_m\right) - n_0^2\right] - 2\frac{\cos^4\rho_j}{\eta^2}\frac{1}{(\Delta\rho)^2} \tag{13}$$

$$C_j = \frac{\cos^4\rho_j}{\eta^2}\frac{1}{(\Delta\rho)^2} + \frac{\cos^3\rho_j}{\eta^2\sin\rho_j}\frac{1}{2(\Delta\rho)^2}\left(2\cos^2\rho_j - 1\right).$$

Substituting Eqs. (11) and (12) into Eq. (9) yields

$$\begin{aligned}
&\beta A_j A_{j-1} H_{j-2}^{m+1} + \left(\alpha A_j + \beta A_j B_{j-1}^{m+1} + \beta A_j B_j^{m+1}\right) H_{j-1}^{m+1} \\
&+ \left(1 + \alpha B_j^{m+1} + \beta A_j C_{j-1} + \beta B_j^{m+1} B_j^{m+1} + \beta A_{j+1} C_j\right) H_j^{m+1} \\
&+ (\alpha C_j + \beta B_j^{m+1} C_j + \beta B_{j+1}^{m+1} C_j) H_{j+1}^{m+1} + \beta C_j C_{j+1} H_{j+2}^{m+1} \\
&= \gamma A_j A_{j-1} H_{j-2}^{m} + \left(\gamma A_j + \delta A_j B_{j-1}^{m} + \delta A_j B_j^{m}\right) H_{j-1}^{m} \\
&+ \left(1 + \gamma B_j^{m} + \delta A_j C_{j-1} + \delta B_j^{m} B_j^{m} + \delta A_{j+1} C_j\right) H_j^{m} \\
&+ \left(\gamma C_j + \delta B_j^{m} C_j + \delta B_{j+1}^{m} C_j\right) H_{j+1}^{m} + \delta C_j C_{j+1} H_{j+2}^{m}
\end{aligned} \quad . \tag{14}$$

It is clear from Eq. (14) that the system is pentadiagonal.

4. Boundary Conditions

Assuming axial symmetry in the radial direction (i.e. $\rho_{-j} = -\rho_j$ and $H_{-j}^m = H_j^m$) we arrive at the following boundary conditions for j = 0 and j = 1:

$$\begin{aligned}
&\left(1 + \alpha B_0^{m+1} + \beta B_0^{m+1} B_0^{m+1} + \frac{2\beta\cos^4\rho_1}{\eta^4(\Delta\rho)^4} - \frac{\beta}{\eta^4(\Delta\rho)^3}\frac{\cos^3\rho_1(2\cos^2\rho_1 - 1)}{\sin\rho_1}\right) H_0^{m+1} \\
&+ \left(\frac{2\alpha}{\eta^2(\Delta\rho)^2} + \frac{2\beta}{\eta^2(\Delta\rho)^2} B_0^{m+1} + \frac{2\beta}{\eta^2(\Delta\rho)^2} B_1^{m+1}\right) H_1^{m+1} \\
&+ \left(\frac{2\beta}{\eta^4(\Delta\rho)^4}\cos^4\rho_1 + \frac{\beta}{\eta^4(\Delta\rho)^3}\frac{\cos^3\rho_1(2\cos^2\rho_1 - 1)}{\sin\rho_1}\right) H_2^{m+1} \\
&= \left(1 + \gamma B_0^{m} + \delta B_0^{m} B_0^{m} + \frac{2\delta\cos^4\rho_1}{\eta^4(\Delta\rho)^4} - \frac{\delta}{\eta^4(\Delta\rho)^3}\frac{\cos^3\rho_1(2\cos^2\rho_1 - 1)}{\sin\rho_1}\right) H_0^{m} \\
&+ \left(\frac{2\gamma}{\eta^2(\Delta\rho)^2} + \frac{2\delta}{\eta^2(\Delta\rho)^2} B_0^{m} + \frac{2\delta}{\eta^2(\Delta\rho)^2} B_1^{m}\right) H_1^{m} \\
&+ \left(\frac{2\delta}{\eta^2(\Delta\rho)^4}\cos^4\rho_1 + \frac{\delta}{\eta^4(\Delta\rho)^3}\frac{\cos^3\rho_1\left(2\cos^2\rho_1 - 1\right)}{\sin\rho_1}\right) H_2^{m}
\end{aligned} \tag{15}$$

and

$$
\begin{aligned}
&\left(\alpha A_1 + \beta A_1 B_0^{m+1} + \beta A_1 B_1^{m+1}\right) H_0^{m+1} \\
&+ \left(1 + \alpha B_1^{m+1} + \beta A_1 \frac{2\cos^4 \rho_1}{\eta^2 (\Delta\rho)^2} + \beta B_1^{m+1} B_1^{m+1} + \beta A_2 C_1\right) H_1^{m+1} \\
&+ \left(\alpha C_1 + \beta B_1^{m+1} C_1 + \beta B_2^{m+1} C_1\right) H_2^{m1} + \beta C_1 C_2 H_3^{m+1} \\
&= \left(\gamma A_1 + \delta A_1 B_0^{m} + \delta A_1 B_1^{m}\right) H_0^{m} \\
&+ \left(1 + \gamma B_1^{m} + \delta A_1 \frac{2\cos^4 \rho_1}{\eta^2 (\Delta\rho)^2} + \delta B_1^{m} B_1^{m} + \delta A_2 C_1\right) H_1^{m} \\
&+ \left(\gamma C_1 + \delta B_1^{m} C_1 + \delta B_2^{m} C_1\right) H_2^{m} + \delta C_1 C_2 H_3^{m}
\end{aligned} \quad . \qquad (16)
$$

5. Computer Program

The finite-difference model described above was implemented in C++ and is included in Appendix A. The pentadiagonal solver is based on LU decomposition as described in Ref. 10. At the time this report was written, the code would compile but not yet run as expected. This is most likely due to a logical error in the code which should be a minor fix.

6. Conclusions

In the report, a finite-difference model for beam propagation within cylindrical geometry is developed using a slowly varying envelope formalism for the scalar Helmholtz equation. A finite-difference computer model for the (2,2) Padé approximant to the Helmholtz operator was used as it is capable of handling wide-angle propagation and refractive index variation, while still maintaining numerical speed and simplicity A transformation from the infinite physical space to a finite computational space was also employed, eliminating the boundary conditions at the edge of the computational window and possible spurious reflections.

This computer model, developed in C++, is able to couple to a thermal model in a time-slicing scheme and could be used to model the time-dependent light distribution, including thermal lensing effects, within the eye as well as the irradiance incident at the retina. Upon verification of the model with experiment, for example by a z-scan, predicted light distribution values within the eye could be used for practical applications. In particular, the predicted irradiance at the retina of laser light incident on the human eye could be used to establish new MPE limits.

7. References

1. Hadley, G.R., “Wide-angle beam propagation using Padé approximant operators”, Optics Letters 17(20), 1426-1428 (1992).

2. Motamedi, M., Welch, A.J., Cheong, W.F., Ghaffari, S.A., and Tan, O.T., “Thermal lensing in biological medium”, IEEE J. Quantum Electron. 24, 693–696 (1988).

3. Lin, W.C., Motamedi, M., and Welch, A.J., “Nonlinear optical behavior in ocular media during laser irradiation”, Appl. Opt. 34, 7979–7985 (1995).

4. Walsh, J.T. and Cummings, J.P., “Effect of the dynamic optical properties of water on midinfrared laser ablation”, Lasers Surg. Med. 15 295–305 (1994).

5. Lin, W.C., Motamedi, M., and Welch, A.J., “Dynamics of tissue optics during laser heating of turbid media”, Appl. Opt. 35(19), 3413–3420 (1995).

6. Vodopyanov, K.L., “Saturation studies of H20 and HDO near 3400 cm-1 using intense picosecond laser pulses”, J. Chem. Phys. 94(8), 5389-5393 (1991).

7. Stolarski, D.J., Stolarski, J., Thomas, R.J., Noojin, G.D., Schuster, K.J., Rockwell, B.A. “Nonlinear absorption studies of melanin”, Proceedings of SPIE-The International Society for Optical Engineering, 4617 (Laser Tissue Interaction XIII: Photochemical, Photothermal, and Photomechanical), 161-171 (2002).

8. Ladouceur, F., “Boundaryless beam propagation”, Optics Letters 21(1), 4-5 (1996).

9. Chung, Y. and Dagli, N., “An assessment of finite difference beam propagation method”, IEEE Journal of Quantum Electron. 26(8), 1335-1339 (1990).

10. Pozrikidis, C., “Numerical Computation in Science and Engineering”, Oxford University Press (1998).

This page intentionally left blank

Appendix A

```
#include<stdio.h>
#include<stdlib.h>
#include<iostream.h>
#include<complex.h>
#include<math.h>
#include<fstream.h>

double get_n(complex<double> r, complex<double> z);

void get_E0(complex<double> rmax, complex<double> delr, int rn, complex<double> soln);

void pentadiagonalsolver(int n , complex<double> g, complex<double>
h,complex<double> d, complex<double> e, complex<double> , complex<double>
b, complex<double> soln);


double get_n(complex<double> r, complex<double> z)
{

        complex<double> del_n;
        double n;
        del_n = 0.00005;
        double lens = 0.0;
        n = ((1.5 + 1.0*(-lens)) + (lens /(1+ real((r)/(del_n))*real((r)/(del_n)))));
        return(n);

}
```

```
void get_E0(complex<double> rmax, complex<double> delr, int rn, complex<double> soln)
{
	int j;
	for(j=0;j< rn;j++)
	{
	soln[j] = 1 * (std::exp(-(pow((( (j)* real(delr)) / real((rmax)*0.1) ),2))) );
	}

}

void pentadiagonalsolver(int n, complex<double> g, complex<double> h,
complex<double> d, complex<double> e, complex<double> f, complex<double> b,
complex<double> soln)
{

	int k;
	complex<double> alpha[n];
	complex<double> gam[n-1];
	complex<double> delta[n-2];
	complex<double> bet[n];
	complex<double> c[n];
	alpha[0] = d[0];
	gam[0] = e[0]/alpha[0];
	delta[0] = f[0] / alpha[0];
	bet[1] = h[1];
	alpha[1] = d[1] - (bet[1] * gam[0]);
	gam[1] = (e[1] - (bet[1] *delta[0]))/alpha[1];
	delta[1] = f[1]/alpha[1];
```

```
for(k=2;k<=n-3;k++)
{
        bet[k] = h[k] - ( g[k]*gam[k-2]);
        alpha[k] = d[k] - (g[k] * delta[k-2]) - (bet[k] * gam[k-1]);
        gam[k] = ( e[k] - (bet[k]* delta[k-1]))/alpha[k];
        delta[k] = f[k]/alpha[k];

}

bet[n-2] = h[n-2] - (g[n-2] * gam[n-4]);
alpha[n-2] = d[n-2] - (g[n-2]*delta[n-4]) - (bet[n-2] * gam[n-3]);
gam[n-2] = ( e[n-2] - (bet[n-2] * delta[n-3]) ) / alpha[n-2];
bet[n-1] = h[n-1] - (g[n-1] * gam[n-3]);
alpha[n-1] = d[n-1] - (g[n-1] * delta[n-3]) - (bet[n-1] * gam[n-2]);
c[0] = b[0] / alpha[0];
c[1] = (b[1] - (bet[1]*c[0]))/alpha[1];

for(k=2;k<n;k++)
{

        c[k] = (b[k] - (g[k]*c[k-2]) - (bet[k]*c[k-1]))/alpha[k];

}

soln[n-1] = c[n-1];

soln[n-2] = c[n-2] - gam[n-2] *soln[n-1];

for(k=n-3;k>=0;k--)
```

```
            {
                        soln[k] = c[k] - (gam[k] * soln[k+1]) - (delta[k] * soln[k+2]);
            }
}


int main()

{

 complex<double> lambda = 0.000001;
 complex<double> k0 = (( 2 * 3.14159 )/ lambda);
 complex<double> nbar = 1.5;
 complex<double> k = k0 * nbar;
 complex<double> rhomax = 0.00001;
 complex<double> zmax = 0.00004;
 complex<double> delz = 0.0000001;
 complex<double> delrho = 0.0000001;
 complex<double> rhon1;
            rhon1 = rhomax / delrho;
            int rhon = real(rhon1) + 1;
 complex<double> i;
           imag(i) = 1;
           real(i) = 0;
 complex<double> zn1;
            zn1 = zmax / delz;
 I          nt zn = real(zn1) + 1;
 complex<double> eta = 1.0;
 complex<double> DLa[rhon];
 complex<double> DLb[rhon];
 complex<double> DLc[rhon];
```

```
complex<double> DLd[rhon];
complex<double> DLe[rhon];
complex<double> DRa[rhon];
complex<double> DRb[rhon];
complex<double> DRc[rhon];

complex<double> DRd[rhon];
complex<double> DRe[rhon];
complex<double> soln[rhon];

int p;
int j;
int m;

complex<double> A[3];
complex<double> B[3];
complex<double> C[3];
complex<double> rho[rhon+1];
complex<double> z[rhon];

rho[0] = 0;
z[0] = 0;

for(p=1;p<rhon;p++)
{

        rho[p] = rho[p-1] + delrho;

}

 for(p=1;p<zn;p++)
```

```
 {

            z[p] = z[p-1] + delz;

}

rho[rhon] = rhomax + delrho; // Extra Point Needed in Loop when j=rhon-1

complex<double> alpha;
complex<double> beta;
complex<double> gama;
complex<double> delta;

alpha = ( (3.0/(4.0 * k * k)) - ( (i * delz)/(4.0 * k) ) );
beta = ((1.0 / ( 16.0 * k * k * k * k)) - ((i * delz )/ (8.0 * k * k * k)) );
gama = ( (3.0/(4.0 * k * k)) + ( (i * delz)/(4.0 * k) ) );
delta = ((1.0 / ( 16.0 * k * k * k * k)) + ((i * delz )/ (8.0 * k * k * k)) );
double n;



get_E0(rhomax,delrho,rhon,soln);

complex<double> *OldE;

OldE = soln;

double output1, output2;
ofstream outputfile;
outputfile.open("output.txt");
```

```
for(j=0;j<rhon;j++)
{
	output1 = real(OldE[j]);
	output2 = imag(OldE[j]);
	outputfile<<output1<<"\t\t"<<output2<<"\n";

}

for(m=1;m<zn;m++)
{

	// Boundary Conditions

	DLa[0] = 0;
	DLb[0] = 0;
	n = get_n(rho[0], z[m]);

	B[0] = ( ((k0 * k0) * ((n * n )-(nbar * nbar) )) - ( ( 2.0 * pow( (cos (real(rho[0]))), 4))/(eta * eta * delrho * delrho ) ));

	n = get_n(rho[1], z[m]);

	B[1] = ( ((k0 * k0) * ((n * n )-(nbar * nbar) )) - ( ( 2.0 * pow( (cos (real(rho[1]))), 4))/(eta * eta * delrho * delrho ) ));

	DLc[0] = (1.0 + (alpha * B[0])+ (beta * B[0] * B[0])+ ( (2.0 * beta * (pow ( ( cos (real(rho[1]))), 4)))/(( pow (eta, 4))*( pow (
delrho , 4 )) ) ) - ( (beta * (pow( (cos (real(rho[1]))), 3) ) * (2.0 * ( pow ((cos(real(rho[1]))) ,3 )) ) - 1.0 )/ ( (pow( eta, 4)) * ( pow (
delrho , 3)) * sin(real(rho[1])) ) ) );

	DLd[0] = ( ((2.0 * alpha)/(eta * eta * delrho * delrho))+ ((2.0 * beta * B[0])/(eta * eta * delrho * delrho))+ ((2.0 * beta *
B[1])/(eta * eta * delrho * delrho)) );
```

```
DLe[0] = ( ( (2.0 * beta * ( pow ((cos (real(rho[1]))),4 )))/( pow(eta, 4)* pow(delrho,4) )) + ( (beta * (pow( (cos(real(rho[1]))), 3) ) * (2.0 * ( pow ((cos(real(rho[1]))) ,3 )) ) - 1.0 )/ ( (pow( eta, 4)) * ( pow ( delrho , 3)) * sin(real(rho[1])) ) ) );

DRa[0] = 0;

DRb[0] = 0;

n = get_n(rho[0], z[m-1]);

B[0] = ( ((k0 * k0) * ((n * n )-(nbar * nbar) )) - ( ( 2.0 * pow( (cos (real(rho[0]))), 4))/(eta * eta * delrho * delrho ) ));

n = get_n(rho[1], z[m-1]);

B[1] = ( ((k0 * k0) * ((n * n )-(nbar * nbar) )) - ( ( 2.0 * pow( (cos(real(rho[1]))), 4))/(eta * eta * delrho * delrho ) ));

DLc[0] = (1.0 + (gama * B[0])+ (delta * B[0] * B[0])+ ( (2.0 * beta * (pow ( ( cos (real(rho[1]))), 4)))/(( pow (eta, 4))*( pow ( delrho , 4 )) ) ) - ( (delta * (pow( (cos (real(rho[1]))), 3) ) * (2.0 * ( pow ((cos(real(rho[1]))) ,3 )) ) - 1.0 )/ ( (pow( eta, 4)) * ( pow ( delrho , 3)) * sin(real(rho[1])) ) ) );

DLd[0] = ( ((2.0 * gama)/(eta * eta * delrho * delrho))+ ((2.0 * delta * B[0])/(eta * eta * delrho * delrho))+ ((2.0 * delta * B[1])/(eta * eta * delrho * delrho)));

DLe[0] = ( ( (2.0 * delta * ( pow ((cos (real(rho[1]))),4 )))/( pow(eta, 4)* pow(delrho,4) )) + ( (delta * (pow( (cos (real(rho[1]))), 3) ) * (2.0 * ( pow ((cos(real(rho[1]))) ,3 )) ) - 1.0 )/ ( (pow( eta, 4)) * ( pow ( delrho , 3)) * sin(real(rho[1])) ) ) );

DLa[1] = 0;

A[0] = ( ( (pow( (cos (real(rho[0])) ), 4))/ (eta * eta) ) * ( 1.0/ (delrho * delrho) ) ) - ( ( ( pow((cos(real(rho[0]))), 3 ) ) / (eta * eta * sin( real(rho[0]))) ) * (1.0/ (2.0 * delrho) ) * ( (2.0 * (pow( (cos (real(rho[0]))) ,2) ) ) - 1.0 ) ) ;
```

A[1] = (((pow((cos (real(rho[1]))), 4))/ (eta * eta)) * (1.0/ (delrho * delrho))) - (((pow((cos(real(rho[1]))), 3)) / (eta * eta * sin(real(rho[1])))) * (1.0/ (2.0 * delrho)) * ((2.0 * (pow((cos (real(rho[1]))) ,2))) - 1.0)) ;

A[2] = (((pow((cos (real(rho[2]))), 4))/ (eta * eta)) * (1.0/ (delrho * delrho))) - (((pow((cos(real(rho[2]))), 3)) / (eta * eta * sin(real(rho[2])))) * (1.0/ (2.0 * delrho)) * ((2.0 * (pow((cos (real(rho[2]))) ,2))) - 1.0)) ;

n = get_n(rho[0], z[m]);

B[0] = (((k0 * k0) * ((n * n)-(nbar * nbar))) - ((2.0 * pow((cos (real(rho[0]))), 4))/(eta * eta * delrho * delrho)));

n = get_n(rho[1], z[m]);

B[1] = (((k0 * k0) * ((n * n)-(nbar * nbar))) - ((2.0 * pow((cos (real(rho[1]))), 4))/(eta * eta * delrho * delrho)));

n = get_n(rho[2], z[m]);

B[2] = (((k0 * k0) * ((n * n)-(nbar * nbar))) - ((2.0 * pow((cos (real(rho[2]))), 4))/(eta * eta * delrho * delrho)));

C[0] = (((pow((cos (real(rho[0]))), 4))/ (eta * eta)) * (1.0/ (delrho * delrho))) + (((pow((cos(real(rho[0]))), 3)) / (eta * eta * sin(real(rho[0])))) * (1.0/ (2.0 * delrho)) * ((2.0 * (pow((cos(real (rho[0]))) ,2))) - 1.0)) ;

C[1] = (((pow((cos (real(rho[1]))), 4))/ (eta * eta)) * (1.0/ (delrho * delrho))) + (((pow((cos(real(rho[1]))), 3)) / (eta * eta * sin(real(rho[1])))) * (1.0/ (2.0 * delrho)) * ((2.0 * (pow((cos (real(rho[1]))) ,2))) - 1.0)) ;

C[2] = (((pow((cos (real(rho[2]))), 4))/ (eta * eta)) * (1.0/ (delrho * delrho))) + (((pow((cos(real(rho[2]))), 3)) / (eta * eta * sin(real(rho[2])))) * (1.0/ (2.0 * delrho)) * ((2.0 * (pow((cos (real(rho[2]))) ,2))) - 1.0)) ;

DLb[1] = ((A[1] * alpha) + (beta * A[1] * B[0]) + (beta * B[1] * A[1]));

DLc[1] = (1.0 + (B[1] * alpha) + ((beta * A[1] * 2.0 * pow((cos(real(rho[1]))),4))/(eta * eta * delrho * delrho)) + (beta * B[1] * B[1]) + (beta * C[1] * A[2]));

```
DLd[1] = ((C[1] * alpha)+ (beta * B[1] * C[1])+ (beta * C[1] * B[2]) );

DLe[1] = (beta * C[1] * C[2]);

n = get_n(rho[0], z[m-1]);

B[0] = ( ((k0 * k0) * ((n * n )-(nbar * nbar) )) - ( ( 2.0 * pow( (cos (real(rho[0]))), 4))/(eta * eta * delrho * delrho ) ));

n = get_n(rho[1], z[m-1]);

B[1] = ( ((k0 * k0) * ((n * n )-(nbar * nbar) )) - ( ( 2.0 * pow( (cos (real(rho[1]))), 4))/(eta * eta * delrho * delrho ) ));

n = get_n(rho[2], z[m-1]);

B[2] = ( ((k0 * k0) * ((n * n )-(nbar * nbar) )) - ( ( 2.0 * pow( (cos (real(rho[2]))), 4))/(eta * eta * delrho * delrho ) ));

DRa[1] = 0;

DRb[1] = ((A[1] * gama) + (delta * A[1] * B[0]) + (delta * B[1] * A[1]) );

DLc[1] = ( 1.0 + (B[1] * gama) + ( ( delta * A[1] * 2.0 * pow( (cos(real(rho[1]))),4))/( eta * eta * delrho * delrho) ) + (delta *
B[1] * B[1]) + ( delta * C[1] * A[2]) );

DRd[1] = ((C[1] * gama) + (delta * B[1] * C[1]) + (delta * C[1] * B[2]) );

DRe[1] = (delta * C[1] * C[2]);

for(j=2;j<rhon;j++)
{
```

```
A[0] = ( ( (pow( (cos (real(rho[j-1])) ), 4))/ (eta * eta) ) * ( 1.0/ (delrho * delrho) ) ) - ( ( ( pow((cos(real(rho[j-1]))), 3 ) ) / (eta * eta * sin(real( rho[j-1]))) ) * (1.0/ (2.0 * delrho) ) * ( (2.0 * (pow( (cos (real(rho[j-1]))) ,2) ) ) - 1.0 ) ) ;

A[1] = ( ( (pow( (cos (real(rho[j] )) ), 4))/ (eta * eta) ) * ( 1.0/ (delrho * delrho) ) ) - ( ( ( pow((cos(real(rho[j] ))), 3 ) ) / (eta * eta * sin(real( rho[j] ))) ) * (1.0/ (2.0 * delrho) ) * ( (2.0 * (pow( (cos (real(rho[j] ))) ,2) ) ) - 1.0 ) );

A[2] = ( ( (pow( (cos (real(rho[j+1])) ), 4))/ (eta * eta) ) * ( 1.0/ (delrho * delrho) ) ) - ( ( ( pow((cos(real(rho[j+1]))), 3 ) ) / (eta * eta * sin(real( rho[j+1]))) ) * (1.0/ (2.0 * delrho) ) * ( (2.0 * (pow( (cos (real(rho[j+1]))) ,2) ) ) - 1.0 ) ) ;

n = get_n(rho[j-1], z[m]);

B[0] = ( ((k0 * k0) * ((n * n )-(nbar * nbar) )) - ( ( 2.0 * pow( (cos (real(rho[j-1]))), 4))/(eta * eta * delrho * delrho ) ));

n = get_n(rho[j], z[m]);

B[1] = ( ((k0 * k0) * ((n * n )-(nbar * nbar) )) - ( ( 2.0 * pow( (cos (real(rho[j] ))), 4))/(eta * eta * delrho * delrho ) ));

n = get_n(rho[j+1], z[m]);

B[2] = ( ((k0 * k0) * ((n * n )-(nbar * nbar) )) - ( ( 2.0 * pow( (cos (real(rho[j+1]))), 4))/(eta * eta * delrho * delrho ) ));

C[0] = ( ( (pow( (cos (real(rho[j-1])) ), 4))/ (eta * eta) ) * ( 1.0/ (delrho * delrho) ) ) + ( ( ( pow((cos(real(rho[j-1]))), 3 ) ) / (eta * eta * sin(real( rho[j-1]))) ) * (1.0/ (2.0 * delrho) ) * ( (2.0 * (pow( (cos (real(rho[j-1]))) ,2) ) ) - 1.0 ) ) ;

C[1] = ( ( (pow( (cos (real(rho[j] )) ), 4))/ (eta * eta) ) * ( 1.0/ (delrho * delrho) ) ) + ( ( ( pow((cos(real(rho[j] ))), 3 ) ) / (eta * eta * sin(real( rho[j] ))) ) * (1.0/ (2.0 * delrho) ) * ( (2.0 * (pow( (cos (real(rho[j] ))) ,2) ) ) - 1.0 ) ) ;

C[2] = ( ( (pow( (cos (real(rho[j+1])) ), 4))/ (eta * eta) ) * ( 1.0/ (delrho * delrho) ) ) + ( ( ( pow((cos(real(rho[j+1]))), 3 ) ) / (eta * eta * sin(real( rho[j+1]))) ) * (1.0/ (2.0 * delrho) ) * ( (2.0 * (pow( (cos (real(rho[j+1]))) ,2) ) ) - 1.0 ) ) ;

DLa[j] = (beta * A[1] * A[0]);
```

```
DLb[j] = ((A[1] * alpha) + (beta * A[1] * B[0]) + (beta * B[1] * A[1]) );

DLc[j] = ( 1.0 + (B[1] * alpha) + (beta * A[1] * C[0]) + (beta * B[1] * B[1]) + (beta * C[1] * A[2]));

DLd[j] = ((C[1] * alpha)+ (beta * B[1] * C[1])+ (beta * C[1] * B[2]) );

DLe[j] = (beta * C[1] * C[2]);

n = get_n(rho[j-1], z[m-1]);

B[0] =( ((k0 * k0) * ((n * n )-(nbar * nbar) )) - ( ( 2.0 * pow( (cos (real(rho[j-1]))), 4))/(eta * eta * delrho * delrho ) ));

n = get_n(rho[j], z[m-1]);

B[1] = ( ((k0 * k0) * ((n * n )-(nbar * nbar) )) - ( ( 2.0 * pow( (cos(real(rho[j] ))), 4))/(eta * eta * delrho * delrho ) ));

n = get_n(rho[j+1], z[m-1]);

B[2] = ( ((k0 * k0) * ((n * n )-(nbar * nbar) )) - ( ( 2.0 * pow( (cos(real(rho[j+1]))), 4))/(eta * eta * delrho * delrho ) ));

DRa[j] = 0;

DRb[j] = ((A[1] * gama) + (delta * A[1] * B[0]) + (delta * B[1] * A[1]) );

DRc[j] = ( 1.0 + (B[1] * gama) + (delta * A[1] * C[0]) + (delta * B[1] * B[1]) + (delta * C[1] * A[2]));

DRd[j] = ((C[1] * gama) + (delta * B[1] * C[1]) + (delta * C[1] * B[2]) );

DRe[j] = (delta * C[1] * C[2]);
```

```
}

DLe[rhon-2] = 0;
DLd[rhon-1] = 0;
DLe[rhon-1] = 0;
DRd[rhon-1] = 0;
DRe[rhon-1] = 0;
DRe[rhon-2] = 0;

// Compute RHS Column Vector

complex<double> D[rhon];
D[0] = ( DRc[0] * OldE[0] ) + ( DRd[0] * OldE[1] ) + ( DRe[0] * OldE[2]);
D[1] = ( DRb[0] * OldE[0] ) + ( DRc[0] * OldE[1] ) + ( DRd[0] * OldE[2]) + (  DRe[0] * OldE[3] );

for(j=2;j<rhon-2;j++)
{

        D[j] = ( DRa[j] * OldE[j-2] ) + ( DRb[j] * OldE[j-1] ) + ( DRc[j] * OldE[j] )+ ( DRd[j] * OldE[j+1] ) + ( DRe[j] * OldE[j+2]
);

 }

 D[rhon-1] = ( DRa[rhon-1] * OldE[rhon-3] ) + ( DRb[rhon-1] * OldE[rhon-2] ) + (
DRc[rhon-1] * OldE[rhon-1] );

D[rhon-2] = ( DRa[rhon-2] * OldE[rhon-4] ) + ( DRb[rhon-2] * OldE[rhon-3] ) + ( DRc[rhon-2] * OldE[rhon-2] ) + ( DRd[rhon-2] *
OldE[rhon-1] );
```

```
// Solve System

 pentadiagonalsolver(rhon, DLa, DLb, DLc, DLd, DLe, D, soln);



OldE = soln;

// Write to Output File

 for(j=0;j<rhon;j++)
 {
            output1 = real(OldE[j]);
            output2 = imag(OldE[j]);
            outputfile<<output1<<"\t\t"<<output2<<"\n";
 }
 }
outputfile.close();
return(0);
}
```